

## REMARKS

Applicant is in receipt of the Office Action mailed December 8, 2005. Claims 1-31 were rejected. New claims 32-36 have been added. Claims 1-36 are currently pending in the application. Reconsideration of the case is earnestly requested in light of the following remarks.

### Section 102 Rejections

The previous rejections under 35 U.S.C. 102(b) based on Kaiser were withdrawn in light of the Request for Pre-Appeal Brief Review filed on August 8, 2005. In the current Office Action, claims 1-31 were rejected under 35 U.S.C. 102(b) as being anticipated by U.S. Patent No. 5,291,587 to Kodosky et al. (hereinafter "Kodosky"). Applicant respectfully traverses these rejections.

Kodosky relates to a graphical programming system in which graphical programs generally execute according to a data flow paradigm (See, for example, the Abstract; Col. 3, lines 15-22; Col. 3, lines 30-35; Col. 9, lines 25-28; Col. 11, lines 7-10; Col. 13, lines 1-7; and Col. 13, lines 51-56). As described at Col. 3, lines 15-22,

A general type of programming technique involves data flow programming. Data flow programming typically involves an ordering of operations which is not specifically specified by a user but which is implied by data interdependencies. An advantage of data flow programming is that it introduces parallelism into a computer system which, of course, usually increases the speed and efficiency of the system.

Thus, Kodosky clearly teaches graphical programs that generally execute according to data flow principles, in which an ordering of operations is not specifically specified by a user but is implied by data interdependencies. For example, FIG. 57 illustrates a graphical program that includes two For Loop structures. The order of execution of the For Loop structures is not explicitly specified. Instead, each of the For Loops begins executing when its data inputs are ready (See Col. 11, lines 7-10). If both of the For Loops receive their respective data inputs at about the same time (as is likely in this example) then the For Loops may execute in parallel with each other (or in a pseudo-parallel multi-threaded fashion).

In contrast, claim 1 recites a method in which a portion of graphical source code is included in each of a plurality of frames in response to user input, where two or more of the frames are visible at the same time, and where the plurality of frames define an execution order for the plurality of portions of graphical source code such that, during execution of the graphical program, the plurality of portions of graphical source code are executed sequentially in accordance with the execution order defined by the plurality of frames. Kodosky simply does not teach these limitations. In particular, Kodosky does not teach a method where a plurality of frames are displayed on a display and where the frames themselves define an execution order for graphical code contained therein. In the cited portion of Kodosky, the data flow wires define the execution order, if any, between the two For loops.

Thus, for at least the reasons set forth above, Applicant respectfully submits that the subject matter recited in claim 1 is patentably distinct over Kodosky. Inasmuch as independent claims 14 and 21 recite similar subject matter as claim 1, Applicant respectfully submits that these claims are also patentably distinct over Kodosky.

Since the independent claims have been shown to be patentably distinct, the dependent claims are also patentably distinct over Kodosky for at least this reason. Applicant also respectfully submits that numerous ones of the dependent claims recite further distinctions over Kodosky. However, since the independent claims have been shown to be patentably distinct, a further discussion of the dependent claims is not necessary at this time.

New claim 32 recites the additional limitation of, “wherein the execution order for the plurality of portions of graphical source code is defined independently of data flow among the plurality of portions of graphical source code.” Kodosky does not teach a plurality of frames which each include a portion of graphical source code, where two or more of the frames are visible at the same time, and where the plurality of frames define an execution order for the plurality of portions of graphical source code independently of data flow among the plurality of portions of graphical source code. Similarly, Kodosky does not teach the limitation of, “wherein no wires among the plurality of portions of graphical source code are required in order to define the execution order for the plurality of portions of graphical source code”, as recited in new claim 33.

New claim 34 recites the additional limitation of, “wherein the plurality of frames provide a clear visual indication of the execution order for the plurality of portions of graphical source code”. Kodosky teaches graphical programs that generally execute according to data flow principles, in which an ordering of operations is implied by data interdependencies. Kodosky does not teach a plurality of frames that define an execution order for portions of graphical source code included in the respective frames and provide a clear visual indication of the execution order for the portions of graphical source code.

New claim 35 recites the additional limitations of, “wherein each frame is displayed side by side in a top-to-bottom order” and “wherein the plurality of frames define an execution order for the plurality of portions of graphical source code such that during execution of the graphical program the plurality of portions of graphical source code are executed sequentially in the top-to-bottom order.” Kodosky teaches nothing at all regarding these limitations.

New claim 36 recites the additional limitations of, “wherein the frames are displayed in a linear order such that adjacent frames appear to contact each other.” Kodosky does not teach this limitation.

**CONCLUSION**


Applicant submits the application is in condition for allowance, and an early notice to that effect is requested.

If any extensions of time (under 37 C.F.R. § 1.136) are necessary to prevent the above referenced application(s) from becoming abandoned, Applicant(s) hereby petition for such extensions. If any fees are due, the Commissioner is authorized to charge said fees to Meyertons, Hood, Kivlin, Kowert & Goetzel PC Deposit Account No. 50-1505/5150-49000/JCH.

Also enclosed herewith are the following items:

☒ Return Receipt Postcard

Respectfully submitted,



---

Jeffrey C. Hood  
Reg. No. 35,198  
Attorney for Applicant

Meyertons, Hood, Kivlin, Kowert & Goetzel PC  
P.O. Box 398  
Austin, TX 78767-0398  
Phone: (512) 853-8800  
Date: \_\_\_\_\_ JCH/JLB